

DODATKOWY OPIS DO: generacji rozkładu prawdopodobieństwa Bernouliego

```
void Prawdopodobienstwo(int n, double p)
{
    int k;
    double wynik;

    printf("Rozkład Bernuliego dla prawdopodobienstwa sukcesu: %.1f wynosi: \n", p);

    for (k = 1; k <= n; k++)
    {
        wynik = rekurencyjnie(n,k) * pow(p,k)*pow(1-p,n-k);
        printf("Dla k=%d - Pn(%d)=%f\n", k,k,wynik);
    }
}
```

Zdefiniowałem funkcję Prawdopodobienstwo z 2 zmiennymi n i p
następnie zdefiniowałem zmienną int k oraz double wynik która przetrzymuje wynik

następnie określona funkcja for od k=1 do k<=n gdzie k++
oblicza wynik na podstawie wzoru:

$$P_n(k) = \binom{n}{k} p^k (1 - p)^{n-k},$$

oblicza wynik. W programie całość wymaga działania funkcji matematycznej pow(x,x) i użycia biblioteki <math.h> oraz przy włączaniu programu wpisania „gcc zad2.c -o zad2 -lm”.

Czyli tak: wynik = funkcja rekurencyjnie() która liczy dwumian newtona* p do potęgi k* 1- p do potęgi n-k.

DZIAŁANIE PROGRAMU:

```
bartosz@Baxing-VB:~/Desktop/C/Zestaw2$ ./zad2 -M1 8 4
Dwumian newtona (z 8 nad 4) = 70 (użyta metoda: 1)

Rozkład Bernuliego dla prawdopodobienstwa sukcesu: 0.5 wynosi:
Dla k=1 - Pn(1)=0.031250
Dla k=2 - Pn(2)=0.109375
Dla k=3 - Pn(3)=0.218750
Dla k=4 - Pn(4)=0.273438
Dla k=5 - Pn(5)=0.218750
Dla k=6 - Pn(6)=0.109375
Dla k=7 - Pn(7)=0.031250
Dla k=8 - Pn(8)=0.003906
```